

09764670-011901  
P0870-02979260

**Patent Application of  
Jeff S. Eder  
for  
A METHOD OF AND SYSTEM FOR  
EVALUATING CASH FLOW AND ELEMENTS OF A BUSINESS ENTERPRISE**

**CONTINUATION IN PART AND  
CROSS REFERENCE TO RELATED PATENT**

This application is a continuation in part of application number 08/779,109 filed January 6, 1997 and application number 08/539,020 filed October 3, 1995 now abandoned. The subject matter of this application is related to the subject matter of U.S. Patent 5,615,109 for "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets", by Jeff S. Eder, the disclosure of which is incorporated herein by reference.

## BACKGROUND OF THE INVENTION

This invention relates to a method of and system for business valuation, more particularly, to an automated system that determines the relative contribution of different elements of value to the total business cash flow and valuation.

The valuation of a business is complex and time-consuming undertaking. Business valuations determine the price that a hypothetical buyer would pay for a business under a given set of circumstances. The volume of business valuations being performed each year is increasing significantly. A leading cause of this growth in volume is the increasing use of mergers and acquisitions as vehicles for corporate growth. Business valuations are frequently used in setting the price for a business that is being bought or sold. Another reason for the growth in the volume of business valuations has been their increasing use in areas other than supporting merger and acquisition transactions. For example, business valuations are now being used by financial institutions to determine the amount of credit that should be extended to a company, by courts in determining litigation settlement amounts and by investors in evaluating the performance of company management.

In most cases, a business valuation is completed by an appraiser or a Certified Public Accountant (hereinafter, appraiser) using a combination of judgment, experience and an understanding of generally accepted valuation principles. The two primary types of business valuations that are widely used and accepted are income valuations and asset valuations. Market valuations are also used in some cases but their use is restricted because of the difficulty inherent in trying to compare two different companies.

Income valuations are based on the premise that the current value of a business is a function of the future value that an investor can expect to receive from purchasing all or part of the business. Income valuations are the most widely used type of valuation. They are generally used for valuing businesses that are expected to continue operating for the foreseeable future. In these valuations the expected returns from investing in the business and the risks associated with receiving the expected returns are evaluated by the appraiser. The appraiser then determines the value whereby a hypothetical buyer would receive a sufficient return on the investment to compensate the buyer for the risk

associated with receiving the expected returns. Income valuation methods include the capitalization of earnings method, the discounted future income method, the discounted cash flow method, the economic income method and other formula methods.

Asset valuations consider the business to be a collection of assets which have an intrinsic value to a third party in an asset sale. Asset valuations are typically used for businesses that are ceasing operation and for specific type of businesses such as holding companies and investment companies. Asset valuation methods include the book value method, the adjusted book value method, the economic balance sheet method and the liquidation method.

Market valuations are used to place a value on one business by using valuations that have been established for comparable businesses in either a public stock market or a recent transaction. This method is difficult to use properly because no two companies are exactly the same and no two transactions are completed for the exact same reasons. Market valuation methods include the price to earnings method, the comparable sales method, industry valuation methods and the comparable investment method.

When performing a business valuation, the appraiser is generally free to select the valuation type and method (or some combination of the methods) in determining the business value. Under the current procedures, there is no correct answer, there is only the best possible informed guess for any given business valuation. There are several difficulties inherent in this approach. First, the reliance on informed guessing places a heavy reliance on the knowledge and experience of the appraiser. The recent increase in the need for business valuations has strained the capacity of existing appraisal organizations. As a result, the average experience level of those performing the valuations has decreased. The situation is even worse for many segments of the American economy where experienced appraisers don't exist because the industries are too new. Another drawback of the current procedures for completing a valuation is that the appraiser is typically retained and paid by a party to a proposed transaction. It is difficult in this situation to be certain that the valuation opinion is unbiased and fair. Given the appraiser's wide latitude for selecting the method, the large variability of experience levels in the industry and the high likelihood of appraiser bias, it is not surprising that it is generally very difficult to compare the valuations of two different appraisers - even for the

same business. These limitations in turn serve to seriously diminish the usefulness of business valuations to business managers, business owners and financial institutions.

The usefulness of business valuations to business owners and managers is limited for another reason - valuations typically determine only the value of the business as a whole. To provide information that would be useful in improving the business, the valuation would have to furnish supporting detail that would highlight the value of different elements of the business. An operating manager would then be able to use a series of business valuations to identify elements within a business that have been decreasing in value. This information could also be used to identify corrective action programs and to track the progress that these programs have made in increasing business value. This same information could also be used to identify elements that are contributing to an increase in business value. This information could be used to identify elements where increased levels of investment would have a significant favorable impact on the overall health of the business.

Another limitation of the current methodology is that financial statements and accounting records have traditionally provided the basis for most business valuations. Appraisers generally spend a great deal of time extracting, aggregating, verifying and interpreting the information from accounting systems as part of the valuation process. Accounting records do have the advantage of being prepared in a generally unbiased manner using the consistent framework of Generally Accepted Accounting Principles (hereinafter, GAAP). Unfortunately, these accounting statements have proved to be increasingly inadequate for use in evaluating the financial performance of modern companies.

Many have noted that traditional accounting systems are driving information-age managers to make the wrong decisions and the wrong investments. Accounting systems are "wrong" for one simple reason, they track tangible assets while ignoring intangible assets. Intangible assets such as the skills of the workers, intellectual property, business infrastructure, databases, and relationships with customers and suppliers are not measured with current accounting systems. This oversight is critical because in the present economy the success of an enterprise is determined more by its ability to use its

intangible assets than by its ability to amass and control the physical ones that are tracked by traditional accounting systems.

The recent experience of several of the most important companies in the U.S. economy, IBM, General Motors and DEC, illustrates the problems that can arise when intangible asset information is omitted from corporate financial statements. All three were all showing large profits using current accounting systems while their businesses were falling apart. If they had been forced to take write-offs when the declines in intangible assets were occurring, the problems would have been visible to the market and management would have been forced to act on them much sooner. These deficiencies of traditional accounting systems are particularly noticeable in high technology companies that are highly valued for their intangible assets and their options to enter new markets rather than their tangible assets.

The accounting profession itself recognizes the limitations of traditional accounting systems. A group of senior financial executives, educators and consultants that had been asked to map the future of financial management by the American Institute of Certified Public Accountants (AICPA) recently concluded that:

- a) Operating managers will continue to lose confidence in traditional financial reporting systems,
- b) The motto of CFOs in the future will likely be "close enough is good enough", and
- c) The traditional financial report will never again be used as the exclusive basis for any business decisions.

The deficiency of traditional accounting systems is also one of the root causes of the short term focus of many American firms. Because traditional accounting methods ignore intangible assets, expenditures that develop a market or expand the capabilities of an organization are generally shown as expenses that only decrease the current period profit. For example, an expenditure for technical training which increases the value of an employee to an enterprise is an expense while an expenditure to refurbish a piece of furniture is capitalized as an asset.

The dependence on accounting records for valuing business enterprises has to some extent been a matter of simple convenience. Because corporations are required to maintain financial records for tax purposes, accounting statements are available for virtually every company. At the same time, the high cost of data storage has until recently prevented the more detailed information required for valuing intangibles from being readily available. In a similar manner, the absence of integrated corporate databases within corporations and the home-grown nature of most corporate systems has until recently made it difficult to compare similar data from different firms.

The lack of a consistent, well accepted, realistic method for measuring all the elements of business value also prevents some firms from receiving the financing they need to grow. Most banks and lending institutions focus on book value when evaluating the credit worthiness of a business seeking funds. As stated previously, the value of many high technology firms lies primarily in intangible assets and growth options that aren't visible under traditional definitions of accounting book value. As a result, these businesses generally aren't eligible to receive capital from traditional lending sources, even though their financial prospects are generally far superior to those of companies with much higher tangible book values.

In light of the preceding discussion, it is clear that it would be advantageous to have an automated financial system that measured the financial performance of all the elements of business value for a given enterprise. Ideally, this system would be capable of generating detailed valuations for businesses in new industries.

## SUMMARY OF THE INVENTION

It is a general object of the present invention to provide a novel and useful system that calculates and displays a comprehensive and accurate valuation for the elements of an enterprise that overcomes the limitations and drawbacks of the prior art that were described previously.

A preferable object to which the present invention is applied is the valuation of elements of a high technology commercial enterprise where a significant portion of the business value is associated with intangible assets.

The present invention eliminates a great deal of time-consuming and expensive effort by automating the extraction of transaction data from the databases, tables, and files of the existing computer-based corporate finance, operation, sales, and human resource software databases as required to operate the system. In accordance with the invention, the automated extraction, aggregation and analysis of transaction data from a variety of existing computer-based systems significantly increases the scale and scope of the analysis that can be completed. The system of the present invention further enhances the efficiency and effectiveness of the business valuation by automating the retrieval, storage and analysis of information useful for valuing intangible assets from external databases and publications via the internet or other external networks.

Uncertainty over which method is being used for completing the valuation and the resulting inability to compare different valuations is eliminated by the present invention by consistently utilizing different valuation methodologies for valuing the different segments of the enterprise as shown in Table 1.

**Table 1**

<b>Segments of Enterprise Value</b>	<b>Valuation methodology</b>
• <b>Excess Cash &amp; Marketable Securities</b>	GAAP
• <b>Total current-operation value (COPTOT):</b>	Income valuation*
Current-operation: Cash & Marketable Securities (CASH)	GAAP
Current-operation: Accounts Receivable (AR)	GAAP
Current-operation: Inventory (IN)	GAAP
Current-operation: Prepaid Expenses (PE)	GAAP
Current-operation: Production Equipment (PEQ)	If correlation value > liquidation value, then use correlation valuation, else use liquidation value
Current-operation: Other Physical Assets (OPA)	Liquidation Value
Current-operation: Other Assets (OA)	GAAP
Current-operation: <u>Elements (E):</u> Customers Employees Vendor Relationships Strategic Partnerships Brand Names Other Elements	Correlation to component(s) of value Correlation to component(s) of value Correlation to component(s) of value Correlation to component(s) of value Correlation to component(s) of value Correlation to component(s) of value
Current-operation: General going concern value (GCV)	GCV = COPTOT - CASH - AR - IN - PE - PEQ - OPA - OA - E

\* The user also has the option of specifying the total value

The value of an enterprise operation is calculated by summing items from Table 1 as shown in Table 2.

**Table 2**

<b>Enterprise Operation Value =</b>
Current value of enterprise excess cash and marketable securities + Value of current-operation



The innovative system has the added benefit of providing a large amount of detailed information concerning both tangible and intangible elements of enterprise business value. The system also gives the user the ability to track the changes in elements of business value and total business value over time by comparing the current valuation to previously calculated valuations. As such, the system also provides the user with an alternative mechanism for tracking financial performance. To facilitate its use as a tool for improving the value of an enterprise, the system of the present invention produces reports in formats that are similar to the reports provided by traditional accounting systems. The method for tracking the elements of value for a business enterprise provided by the present invention eliminates many of the limitations associated with current accounting systems that were described previously.

### **BRIEF DESCRIPTION OF DRAWINGS**

These and other objects, features and advantages of the present invention will be more readily apparent from the following description of the preferred embodiment of the invention in which:

FIG. 1 is a block diagram showing the major processing steps of the present invention;

FIG. 2 is a diagram showing the files or tables in the application database of the present invention that are utilized for data storage and retrieval during the processing that values elements of the enterprise;

FIG. 3 is a block diagram of an implementation of the present invention;

FIG. 4 is a diagram showing the data windows that are used for receiving information from and transmitting information to the user during system processing;

FIG. 5A and FIG. 5B are block diagrams showing the sequence of steps in the present invention used for extracting, aggregating and storing information utilized in system processing from: user input, the basic financial system database, the operation management system database, the advanced financial system database, the sales management system database, external databases via the internet and the human resource information system database;

FIG. 6 is a block diagram showing the sequence of steps associated with the calculation of the composite variables that characterize the performance of the elements of value;

FIG. 7 is a block diagram showing the sequence of steps associated with the calculation of the cash flow and components of enterprise value;

FIG. 8A, FIG. 8B, FIG. 8C and FIG. 8D are block diagrams showing the sequence of steps in the present invention that are utilized in the specification and optimization of the predictive models that determine the relationships between elements (and sub-elements) of value and the revenue component, expense component, capital component and cash flow of the enterprise;

FIG. 9 is a diagram illustrating the processing of a feed-forward neural network;

FIG. 10 is a diagram illustrating the processing of a Kohonen neural network;

FIG. 11A and FIG. 11B are block diagrams showing the sequence of the steps in the present invention used for calculating the percentage of the revenue component, expense components, capital components and cash flow attributed to the elements and sub-elements of value;

FIG. 12 is a block diagram showing the sequence of steps in the present invention used in preparing, displaying and optionally printing reports

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

FIG.1 provides an overview of the processing completed by the innovative system for business valuation. In accordance with the present invention, an automated method of and system (100) for business valuation is provided. Processing starts in this system (100) with a block of software (200) that extracts, aggregates and stores the transaction data and user input required for completing a valuation. This information is extracted via an interconnection network (25) from a basic financial system database (10), an operation management system database (15), an advanced financial system database (30), a sales management system database (35), and a human resource information system database (40). Information can also be extracted from an on-line external database such as those found on an internet (5) via a communications link (45). These information extractions and aggregations are guided by a user (20) through interaction with a user-interface portion of the application software (900) that mediates the display and transmission of all information to the user (20) from the system (100) as well as the receipt of information into the system (100) from the user (20) using a variety of data windows tailored to the specific information being requested or displayed in a manner that is well known. While only one database of each type (10, 15, 30, 35 & 40) is shown in FIG. 1, it is to be understood that the system (100) can extract data from multiple databases of each type via the interconnection network (25).

All extracted information concerning revenue, expenses, capital and elements of value is stored in a file or table (hereinafter, table) within an application database (50) as shown in FIG. 2. The application database (50) contains tables for storing user input, extracted information and system calculations including a system settings table (140), a revenue data table (141), an expense data table (142), a capital data table (143), an equity data table (144), a physical asset ID table (145), an asset liquidation price table (146), an account number structure table (147), an equity forecast table (148), a data dictionary table (149), a revenue component definition table (150), an expense component definition table (151), a capital component definition table (152), an element of value definition table (153), a sub-element definition table (154), an enterprise definition table (155), a composite variable table (156), a sub-element weights table (157), a revenue model gene table (158), a revenue model weights table (159), an expense model gene table (160), an expense model weights table (161), a capital model gene table (162), a capital model weights table (163), a revenue component percentage table (164), an expense component percentage table (165), a capital component percentage table (166), a composite variable location table (167), a composite variable data table (168), a normalized composite variable data table (169), an enterprise value table (170), an economic equity values table (171), a reports table (172), a tax data table (173), a debt data table (174), a cash flow model weight table (175), a cash flow model genes table (176) and a cash flow model percentage table (177). The application database (50) can optionally exist as a datamart, data warehouse or departmental warehouse. The system of the present invention has the ability to accept and store supplemental or primary data directly from user input, a data warehouse or other electronic files in addition to receiving data from the databases described previously. The system of the present invention also has the ability to complete the necessary calculations without receiving data from one or more of the specified databases. However, in the preferred embodiment all required information is obtained from the specified databases (5, 10, 15, 30, 35 & 40).

As shown in FIG. 3, the preferred embodiment of the present invention is a computer system (100) illustratively comprised of a client personal computer (110) connected to an application server personal computer (120) via an interconnection network (25). The application server personal computer (120) is in turn connected via the interconnection network (25) to a database-server personal computer (130).

The database-server personal computer (130) has a CPU (136), a keyboard (132), a CRT display (133), a printer (137), a hard drive (131) for storage of the basic financial system database (10), the operation management system database (15), the advanced financial system database (30), the sales management system database (35) and the human resource information system database (40), a communications bus (134) and a read/write random access memory (135).

The application-server personal computer (120) has a CPU (127), a keyboard (123), a mouse (126), a CRT display (124), a printer (128), a hard drive (122) for storage of the application database (50) and the majority of the application software (200, 300, 400, 500, 600, 700) of the present invention, a communications bus (125) and a read/write random access memory (121). While only one client personal computer is shown in FIG. 3, it is to be understood that the application-server personal computer (120) can be networked to fifty or more client personal computers (110) via the interconnection network (25). The application-server personal computer (120) can also be networked to fifty or more server, personal computers (130) via the interconnection network (25). It is to be understood that the diagram of FIG. 3 is merely illustrative of one embodiment of the present invention.

The client personal computer (110) has a CPU (117), a keyboard (113), a mouse (116), a CRT display (114), a printer (118), a modem (119), a hard drive (112) for storage of a client data-base (49) and the user-interface portion of the application software (900), a communications bus (115) and a read/write random access memory (111).

The application software (200, 300, 400, 500, 600, 700 and 900) controls the performance of the central processing unit (127) as it completes the calculations required to calculate the detailed business valuation. In the embodiment illustrated herein, the application software program (200, 300, 400, 500, 600, 700 and 900) is written in a combination of PowerScript, C++ and Visual Basic®. The application software (200, 300, 400, 500, 600, 700 and 900) also uses Structured Query Language (SQL) for extracting data from other databases (10, 15, 30, 35 and 40) and then storing the data in the application database (50) or for receiving input from the user (20) and storing it in the client database (49). The other databases contain information regarding historical financial performance (10), operation management records (15), forecast financial

performance (30), sales prospects and transactions (35) and the company employees (40) that are used in the operation of the system (100). The user (20) provides the information the application software requires to determine which data need to be extracted and transferred from the database-server hard drive (131) via the interconnection network (25) to the application-server computer hard drive (122) by interacting with user-interface portion of the application software (900). The extracted information is combined with input received from the keyboard (113) or mouse (116) in response to prompts from the user-interface portion of the application software (900) before processing is completed.

User input is initially saved to the client database (49) before being transmitted to the communication bus (125) and on to the hard drive (122) of the application-server computer via the interconnection network (25). Following the program instructions of the application software, the central processing unit (127) accesses the extracted data and user input by retrieving it from the hard drive (122) using the random access memory (121) as computation workspace in a manner that is well known.

The computers (110, 120 and 130) shown in FIG. 3 illustratively are IBM-PCs or clones or any of the more powerful computers or workstations that are widely available. Typical memory configurations for client personal computers (110) used with the present invention should include at least 32 megabytes of semiconductor random access memory (111) and at least a 2 gigabyte hard drive (112). Typical memory configurations for the application-server personal computer (120) used with the present invention should include at least 64 megabytes of semiconductor random access memory (121) and at least a 50 gigabyte hard drive (122). Typical memory configurations for the database-server personal computer (130) used with the present invention should include at least 128 megabytes of semiconductor random access memory (135) and at least a 200 gigabyte hard drive (131).

As shown in FIG. 4, various data windows that are used in the user-interface portion of the application software (900) are illustrated. The data windows are used by the present invention for receiving information from and transmitting information to the user (20) during system processing. As will be discussed, data windows utilized by the present invention include System Settings (901), Account Structure and Data Dictionary (902),

Enterprise Definition (903), Component Definition (904), Edit Component Definition (905), Missing Financial Data (906), Element of Value Definition (907), Enter Missing Data (908), Report (909), Tax Information (910), Equity Information (911), Liability Information (912), Cost of Capital Selection (913), Liquidation Price Entry (914) and Report Selection (915).

Using the system described above, the value of the enterprise will be further broken down into tangible and intangible elements of value. As shown in Table 1, the value of the current-operation will be calculated using an income valuation model unless the user (20) decides to specify the current operation value. An integral part of most income valuation models is the calculation of the present value of the expected cash flows, income or profits associated with the current-operation. The present value of a stream of cash flows is calculated by discounting the cash flows at a rate that reflects the risk associated with realizing the cash flow. For example, the present value (PV) of a cash flow of ten dollars (\$10) per year for five (5) years would vary depending on the rate used for discounting future cash flows as shown below.

Discount rate = 25%												
PV	=	$\frac{10}{1.25}$	+	$\frac{10}{(1.25)^2}$	+	$\frac{10}{(1.25)^3}$	+	$\frac{10}{(1.25)^4}$	+	$\frac{10}{(1.25)^5}$	=	26.89

Discount rate = 35%												
PV	=	$\frac{10}{1.35}$	+	$\frac{10}{(1.35)^2}$	+	$\frac{10}{(1.35)^3}$	+	$\frac{10}{(1.35)^4}$	+	$\frac{10}{(1.35)^5}$	=	22.20

The first step in evaluating the elements of current-operation value is separating the underlying formula that defines the value of the current-operation as shown in Table 3.

**Table 3**

<b>Value of current-operation =</b>
(R) Value of expected revenue from current-operation
<b>+</b>
(E) Value of expected expense for current-operation
<b>+</b>
(C) Value of capital required to support current-operation*

\*Note: (C) can have a positive or negative value

The three components of current-operation value will be referred to as the revenue value (R), the expense value (E) and the capital value (C). Examination of the equation in Table 3 shows that there are three ways to increase the value of the current-operation - increase the revenue, decrease the expense or decrease the capital requirements (note: this statement ignores a fourth way to increase value - decrease interest rate used for discounting future cash flows).

While it is possible to break each component down into a large number of sub-components for analysis, the preferred embodiment has a pre-determined number of sub-components for each component of value. Because each enterprise is defined by the revenue stream, there are no sub-components of revenue value. The expense value is subdivided into five sub-components: the cost of raw materials, the cost of manufacture or delivery of service, the cost of selling, the cost of support and the cost of administration. The capital value is subdivided into six sub-components: cash, non-cash financial assets, production equipment, other assets (non financial, non production assets), financial liabilities and equity. The production equipment and equity sub-components are not used directly in evaluating the elements of value.

The components and sub-components of current-operation value will be used in calculating the value of the tangible and intangible elements of value. For the calculations completed by the present invention, a transaction will be defined as any event that is logged or recorded. An element of value will be defined as "an entity or group that as a result of past transactions, forecasts or other data has provided and/or is expected to provide economic benefit to the enterprise." An item will be defined as a single member of the group that defines an element of value. For example, an individual salesman would be an "item" in the "element of value" sales staff. Predictive models are used to

00761670 0297260

determine the percentage of: the revenue value, the expense value sub-components, and the capital value sub-components that are attributable to each element of value. The resulting values will then be added together to determine the valuation for different elements as shown by the example in Table 4.

**Table 4**

<b>Valuation of the Large, Loyal Customer Element</b>		
Revenue value = \$120M	13% attributed to large, loyal customers	Value = \$15.6 M
Expense value = (\$80M)	10% attributed to large, loyal customers	Value = (\$8) M
Capital value = (\$5M)	12% attributed to large, loyal customers	Value = (\$6) M
Total value = \$35M	Large, Loyal Customer Element Value = \$ 7 M	

The system of the present invention also supports an analysis of “cash flow” drivers rather than component drivers which eliminates the need to add the different values together as shown in Table 4. In the preferred embodiment the analysis of cash flow drivers is completed only if the user (20) chooses to run the summary (cash flow) analysis in place of the more detailed analysis of the 3 primary components of any cash flow calculation (revenue, expense and change in capital). The discounted value of projected cash flow used in calculating element values based on cash flow will hereinafter be referred to as the cash flow value.

The valuation of an enterprise using the approach outlined above is completed in six distinct stages. The first stage of processing (block 200 from FIG. 1) extracts, aggregates and stores the data from user input, existing internal databases (10, 15, 30, 35 or 40) and external databases (5) required for the calculation of enterprise business value as shown in FIG. 5A and FIG. 5B. The second stage of processing (block 300 from FIG. 1) calculates composite variables that characterize the performance of the elements of value and optionally creates sub-elements as shown in FIG. 6 and FIG. 10. The third stage of system processing (block 400 from FIG. 1) calculates the cash flow revenue, expense and capital values and optionally the current operation value using the information prepared in the previous stage of processing as shown in FIG. 7. The fourth stage of system processing (block 500 from FIG. 1) specifies and optimizes predictive models to determine the relationship between the elements of value and the cash flow value or the revenue, expense and capital values as shown in FIG. 8A, FIG. 8B and FIG.



9. The fifth stage of processing (block 600 from FIG. 1) combines the results of the third and fourth stages of processing to determine the value of each element as shown in FIG. 11.
11. The sixth and final stage of processing (block 700 from FIG. 1) determines the relationship between equity and calculated total value as shown in FIG. 12 and displays the results of the prior calculations in specified formats.

## **EXTRACTION AND AGGREGATION OF DATA**

The flow diagrams in FIG. 5A and FIG. 5B detail the processing that is completed by the portion of the application software (200) that extracts, aggregates and stores the information required for system operation from: the basic financial system database (10), operation management system database (15), advanced financial system database (30), sales management system database (35), human resource information system database (40), external databases found on the internet (5) and the user (20). A brief overview of the different databases will be presented before reviewing each step of processing completed by this portion (200) of the application software.

Corporate financial software systems are generally divided into two categories, basic and advanced. Advanced financial systems utilize information from the basic financial systems to perform financial analysis, financial planning and financial reporting functions. Virtually every commercial enterprise uses some type of basic financial system as they are required to use these systems to maintain books and records for income tax purposes. An increasingly large percentage of these basic financial systems are resident in microcomputer and workstation systems. Basic financial systems include general-ledger accounting systems with associated accounts receivable, accounts payable, capital asset, inventory, invoicing, payroll and purchasing subsystems. These systems incorporate worksheets, files, tables and databases. These databases, tables and files contain information about the company operations and its related accounting transactions. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. The system is also capable of extracting the required information from a data warehouse (or datamart) when the required information has been pre-loaded into the warehouse.

General ledger accounting systems generally store only valid accounting transactions. As is well known, valid accounting transactions consist of a debit component and a credit component where the absolute value of the debit component is equal to the absolute value of the credit component. The debits and the credits are posted to the separate accounts maintained within the accounting system. Every basic accounting system has several different types of accounts. The effect that the posted debits and credits have on the different accounts depends on the account type as shown in Table 5.

**Table 5**

<b>Account Type:</b>	<b>Debit Impact:</b>	<b>Credit Impact:</b>
Asset	Increase	Decrease
Revenue	Decrease	Increase
Expense	Increase	Decrease
Liability	Decrease	Increase
Equity	Decrease	Increase

General ledger accounting systems also require that the asset account balances equal the sum of the liability account balances and equity account balances at all times.

The general ledger system generally maintains summary, dollar only transaction histories and balances for all accounts while the associated subsystems, accounts payable, accounts receivable, inventory, invoicing, payroll and purchasing, maintain more detailed historical transaction data and balances for their respective accounts. It is common practice for each subsystem to maintain the detailed information shown in Table 6 for each transaction.

**Table 6**

<b>Subsystem</b>	<b>Detailed Information</b>
Accounts Payable	Vendor, Item(s), Transaction Date, Amount Owed, Due Date, Account Number
Accounts Receivable	Customer, Transaction Date, Product Sold, Quantity, Price, Amount Due, Terms, Due Date, Account Number
Capital Asset	Asset ID, Asset Type, Date of Purchase, Purchase Price, Useful Life, Depreciation Schedule, Salvage Value
Inventory	Item Number, Transaction Date, Transaction Type, Transaction Qty, Location, Account Number
Invoicing	Customer Name, Transaction Date, Item(s) Sold, Amount Due, Due Date, Account Number
Payroll	Employee Name, Employee Title, Pay Frequency, Pay Rate, Account Number
Purchasing	Vendor, Item(s), Purchase Quantity, Purchase Price(s), Due Date, Account Number

As is well known, the output from a general ledger system includes income statements, balance sheets and cash flow statements in well defined formats which assist management in measuring the financial performance of the firm during the prior periods when data input have been completed.

Advanced financial systems, including financial planning systems, generally use the same format used by basic financial systems in forecasting income statements, balance sheets and cash flow statements for future periods. Management uses the output from financial planning systems to highlight future financial difficulties with a lead time sufficient to permit effective corrective action and to identify problems in company operations that may be reducing the profitability of the business below desired levels. These systems are most often developed by individuals within companies using 2 and 3 dimensional spreadsheets such as Lotus 1-2-3®, Microsoft Excel® and Quattro Pro®. In some cases, financial planning systems are built within an executive information system (EIS) or decision support system (DSS). For the preferred embodiment of the present invention, the advanced financial system database is the financial planning system database detailed in U.S. Patent 5,165,109 for "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets", by Jeff S. Eder, the disclosure of which is incorporated herein by reference.

While advanced financial systems are similar between firms, operation management systems vary widely depending on the type of company they are supporting. These systems typically have the ability to not only track historical transactions but to forecast future performance. For manufacturing firms, operation management systems such as Enterprise Requirements Planning Systems (ERP), Material Requirement Planning Systems (MRP), Purchasing Systems, Scheduling Systems and Quality Control Systems are used to monitor, coordinate, track and plan the transformation of materials and labor into products. These systems will generally track information about the performance of the different vendors that supply materials to the firm including the information shown in Table 7.

**Table 7**

<b>Operation Management System - Vendor Information</b>	
1. Vendor Name	8. Compliance with ISO 9000
2. Vendor Number	9. Actual lead time required for purchases
3. Commodity Code(s)	10. Terms and conditions for purchases
4. Year to date dollar volume	11. Average Delivery Quantity Variance
5. Historical dollar volume	12. Average Delivery Date Variance
6. Percentage of deliveries rejected by QC	13. EDI* vendor - Yes or No
7. Percentage of deliveries accepted out of specification	*EDI = Electronic Data Interchange

Systems similar to the one described above may also be useful for distributors to use in monitoring the flow of products from a manufacturer.

Operation Management Systems in manufacturing firms may also monitor information relating to the production rates and the performance of individual production workers, production lines, work centers, production teams and pieces of production equipment including the information shown in Table 8.

**Table 8**

<b>Operation Management System - Production Information</b>	
1. ID number (employee id/machine id)	10. Cumulative training time
2. Actual hours - last batch	11. Job(s) certifications
3. Standard hours - last batch	12. Actual scrap - last batch
4. Actual hours - year to date	13. Scrap allowance - last batch
5. Actual/Standard hours - year to date %	14. Actual scrap/allowance - year to date
6. Actual setup time - last batch	15. Rework time/unit last batch
7. Standard setup time - last batch	16. Rework time/unit year to date
8. Actual setup hours - year to date	17. QC rejection rate - batch
9. Actual/Standard setup hrs - yr to date %	18. QC rejection rate - year to date

Operation management systems are also useful for tracking requests for service to repair equipment in the field or in a centralized repair facility. Such systems generally store information similar to that shown below in Table 9.

**Table 9**

<b>Operation Management System - Service Call Information</b>	
1. Customer name	11. Promised type of response
2. Customer number	12. Time person dispatched to call
3. Contract number	13. Name of person handling call
4. Service call number	14. Time of arrival on site
5. Time call received	15. Time of repair completion
6. Product(s) being fixed	16. Actual response type
7. Serial number of equipment	17. Part(s) replaced
8. Name of person placing call	18. Part(s) repaired
9. Name of person accepting call	19. 2nd call required
10. Promised response time	20. 2nd call number

Sales management systems are similar to operation management systems in that they vary considerably depending on the type of firm they are supporting and they generally have the ability to forecast future events as well as track historical occurrences. In firms that sell customized products, the sales management system is generally integrated with an estimating system that tracks the flow of estimates into quotations, orders and eventually bills of lading and invoices. In other firms that sell more standardized products, sales management systems generally are used to track the sales process from lead

generation to lead qualification to sales call to proposal to acceptance (or rejection) and delivery. All sales management systems would be expected to store information similar to that shown below in Table 10.

**Table 10**

<b>Sales Management System - Information</b>	
1. Customer/Potential customer name	9. Sales call history
2. Customer number	10. Sales contact history
3. Address	11. Sales history: product/qty/price
4. Phone number	12. Quotations: product/qty/price
5. Source of lead	13. Custom product percentage
6. Date of first purchase	14. Payment history
7. Date of last purchase	15. Current A/R balance
8. Last sales call/contact	16. Average days to pay

Computer based human resource systems are increasingly used for storing and maintaining corporate records concerning active employees in sales, operations and the other functional specialties that exist within a modern corporation. Storing records in a centralized system facilitates timely, accurate reporting of overall manpower statistics to the corporate management groups and the various government agencies that require periodic updates. In some cases human resource systems include the company payroll system as a subsystem. In the preferred embodiment of the present invention, the payroll system is part of the basic financial system. These systems can also be used for detailed planning regarding future manpower requirements. Human resource systems typically incorporate worksheets, files, tables and databases that contain information about the current and future employees. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. It is common practice for human resource systems to store the information shown in Table 11 for each employee.

**Table 11**

<b>Human Resource System Information</b>	
1. Employee name	10. Employee start date - company
2. Job title	11. Employee start date - department
3. Job code	12. Employee start date - current job
4. Rating	13. Training courses completed
5. Division	14. Cumulative training expenditures
6. Department	15. Salary history
7. Employee No./(Social Security Number)	16. Current salary
8. Year to date - hours paid	17. Educational background
9. Year to date - hours worked	18. Current supervisor

External databases such as those found on the internet (5) can be used for obtaining information that enables the categorization and valuation of assets such as brand names, trademarks and service marks (hereinafter, referred to as brand names). In some cases it can also be used to supplement information obtained from the other databases (10, 15, 30, 35 and 40) that are used in categorizing and evaluating employee groups and other elements of value. In the system of the present invention, the retrieval of information from the internet (5) can be:

- a) targeted to specific on-line publications that provide information relevant to the element being evaluated,
- b) restricted to a simple count of the number of matches a specific trademark generates when entered into a general purpose internet search-engine such as Yahoo!, Lycos, AltaVista or HotBot, or WebCrawler, and
- c) specific searches using commercially available software agents to determine both the number and the type of references (favorable, unfavorable or information only) that have been made concerning a specific trademark in all discovered references.

System processing of the information from the different databases (5, 10, 15, 30, 35 and 40) described above starts in a block 201, FIG. 5A, which immediately passes

processing to a software block 202. The software in block 202 prompts the user via the system settings data window (901) to provide system setting information. The system setting information entered by the user (20) is transmitted via the interconnection network (25) back to the application server (120) where it is stored in the system settings table (140) in the application database (50) in a manner that is well known. The specific inputs the user (20) is asked to provide at this point in processing are shown in Table 12.

**Table 12**

<b>System Settings</b>
1. Mode of operation - stand-alone valuation or comparison to previous valuation
2. Date of business valuation calculation
3. Date of previous valuation (if any)
4. Location (address) of basic financial system data dictionary and data
5. Location (address) of advanced financial system data dictionary and data
6. Location (address) of human resource information system data dictionary and data
7. Location (address) of operation management system data dictionary and data
8. Location (address) of sales management system data dictionary and data
9. Location (address) of any external databases used in the valuation calculation
10. The maximum acceptable age of a valuation (in days)
11. The maximum number of generations to be processed without improving fitness
12. Base currency
13. Currency conversions for any non-base currencies used in the financial systems
14. Weighted average cost of capital (to be used in discounting cash flows)
15. Simplified analysis (no sub-components of expense or capital value)
16. Number of months a product is considered new after it is first produced
17. Amount of cash and marketable securities required for day to day operations
18. Detailed component (revenue, expense, capital change) or summary (cash flow) analysis?

The application of these system settings will be explained as part of the detailed explanation of the system operation.



The software in block 202 uses the valuation date specified by the user (20) to determine the time periods (months) that require data in order to complete the valuation of the current operation and the growth options and stores the resulting date range in the system settings table (140). The valuation of the current operation by the system requires sales, operation, finance, external database and human resource data for the three year period before and the four year period after the specified valuation date. The same information is required even if the user (20) chooses to specify the current operation value. Because of the difficulties inherent in forecasting from the perspective of the past or the future, the specified valuation date is generally within a month of the current system date. After the storage of system setting data is complete, processing advances to a software block 203 where the data dictionaries from the basic financial system database (10), the operation management system database (15), the advanced financial system database (30), the sales management system database (35) and the human resource information system database (40) are extracted and saved in the data dictionary table (149) in the application database (50) and processing advances to a software block 204.

The software in block 204 checks the system settings table (140) in the application database (50) to determine if the current calculation is a comparison to a prior valuation or if it is a stand-alone calculation. If the calculation involves a comparison with a prior valuation, then the software in block 204 retrieves the previously defined account structure, data definitions, enterprise definitions and component definitions and saves them in the appropriate tables for use in the current calculation before processing advances to a software block 209. Alternatively, if the calculation is a stand-alone, then processing advances to a software block 205.

The software in block 205 interacts with an account structure and data dictionary data window (902) that prompts the user for any input that is required to define data fields for the extracted data dictionaries and the data dictionary of the application software of the present invention. This input is also saved to the data dictionary table (149). The software in block 205 also prompts the user (20) via the account structure and data dictionary data window (902) for information that edits or defines the account structure used in the financial system databases. It is common practice for account numbers to

have several segments where each segment represents a different set of subgroups as shown below in Table 13.

**Table 13**

<b>Account Number</b>	<b>01 -</b>	<b>800 -</b>	<b>901 -</b>	<b>677-</b>	<b>003</b>
<b>Segment</b>	<b>Company</b>	<b>Division</b>	<b>Department</b>	<b>Account</b>	<b>Sub-account</b>
<b>Subgroup</b>	<b>Products</b>	<b>Workstation</b>	<b>Marketing</b>	<b>Labor</b>	<b>P.R.</b>
<b>Position</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>

As will be detailed below, the different account number segments are used for separating the financial information for analysis.

After the account structure information is stored in the account number structure table (147) in the application database (50), processing advances to a block 206 where the software in the block interacts with an enterprise definition data window (903) to prompt the user (20) to specify the account number segment or segments that will be used to define the enterprise being valued by the innovative system of the present invention. For example, the user (20) could specify that each division is to be analyzed as a separate enterprise. In this case, if the total company had two business units with six divisions, then the user could specify up to six enterprises as shown in Table 14.

**Table 14**

<b>Products Business Unit</b>	<b>Software Business Unit</b>
1. PC Division	5. Application Software Division
2. Workstation Division	6. Operating System Software Division
3. Mainframe Division	
4. Peripherals Division	

The specified enterprises are then displayed to the user (20) by the software in block 206 via the enterprise definition data window (903). At this point, the user (20) is given the option of combining the enterprises or leaving them in the initial configuration. For example, the user (20) could combine the Personal Computer Product enterprise and the Workstation Product enterprise into one enterprise for the business valuation calculation.

After the enterprise definitions are stored, processing advances to a software block 207 where the software in the block prompts the user (20) via a component definition data window (904) to specify the account segment or segments that will be used to define the expense and capital sub-components for each enterprise. Only account segments with position numbers below those of the segment used for enterprise specification can be used for expense and capital sub-component specification. Continuing the example shown above for a valuation calculation, departments, accounts and sub-accounts are the only segments that can be utilized for expense or capital component and sub-component specification. This limitation is applicable because their position numbers 3, 2 and 1 respectively are below 4, the position number of the division segment that was the lowest position used to define the enterprise. As discussed previously, there is only one revenue component per enterprise; therefore, the enterprise definition automatically defines the revenue component.

For the normal analysis, each enterprise has: one revenue component, five expense sub-components (cost of raw materials, the cost of manufacture or delivery of service, the cost of sales, the cost of support and the cost of administration), four capital sub-components used in the valuation calculation (cash, non-cash financial assets, other (non-financial, non-production) assets, liabilities), and two capital sub-components that are not used directly in the valuation calculation (production equipment and equity). The software in block 207 via the component definition data window (904) will accept all logical combinations of account number segment specifications for a sub-component while preventing the reuse of the same segment for more than one sub-component specification in each enterprise. Sub-component definitions are required even if the user (20) has chosen to run a simplified analysis (i.e., one without sub-components). Table 15 provides examples of expense and capital sub-component definitions.

**Table 15**

<b>Sub-component</b>	<b>Definition</b>
Expense: Cost of materials	Departments 10 - 18, accounts 500 to 505
Expense: Cost of manufacturing	Departments 10 - 18, accounts 506 to 999
Expense: Cost of sales	Department 21, accounts 500 to 999
Capital: Cash	Account 100, all departments
Capital: Liabilities	Accounts 200 - 299, all departments

The software in block 207 saves the new or updated revenue component definitions to the revenue component definition table (150), expense sub-component definitions to the expense component definition table (151) and capital sub-component definitions to the capital component definition table (152). The production equipment and other asset definitions are also used to populate the physical asset ID table (145) and the asset liquidation price table (146) with the names of all assets used by all enterprises.

After the definitions for the revenue, expense and capital components have been stored in the application database (50), processing advances to a software block 209. Processing can also advance to block 209 directly from block 204 if the calculation is a comparison to a prior valuation. The software in block 209 checks to determine if all the available financial data have been included in a revenue, expense, or capital component or sub-component. In the example shown above, block 209 would check to determine that the financial data for all divisions, departments, account numbers and sub-account numbers have been assigned to a component. If the software in block 209 determines that all financial data have been assigned to a component, then processing advances to a software block 210. Alternatively, if the software in block 209 determines that some financial data have not been assigned to a component, then processing advances to a software block 208. The software in block 208 prompts an edit component definition data window (905) to display a screen that provides the user (20) with the ability to redefine previously stored component and sub-component definitions to include the unassigned financial data. The revised component definition(s) are then saved in the appropriate definition table(s) (150, 151 or 152) in the application database (50) and processing returns to block 209 and from there to software block 210.

The software in block 210 retrieves the debit or credit balances from the basic financial system database (10) and the advanced financial system database (30) in account segment position order, lowest position to highest position, for the revenue, expense and capital components for the time periods determined by the software in block 202 and stored in the system settings table (140). Continuing the example, the software in block 210 would first retrieve and total debits and credits in each required period for the sub-components that have sub-account specifications. The higher level specifications, account number, department and division, are observed when data are retrieved for the sub-components with sub-account specifications. The software in block 210 would then retrieve the required data for the sub-components with account number specifications. The higher level specifications, department and division, are observed when data are retrieved for the sub-components with account number specifications. The software in block 210 would finally retrieve the required data for the sub-components with department number specifications. The higher level specification, division, is observed when data are retrieved for these sub-components. This same procedure is completed for each enterprise and the resulting totals are then saved in the appropriate data tables (141-revenue, 142-expense and 143-capital) in the application database (50).

After all the financial data have been extracted and stored in the application database (50), system processing advances to a software block 212. The software in block 212 determines if any of the components or sub-components are missing data for any of the required periods. Missing data is defined as the condition when there is a null value for a sub-component financial data field in a required period. If the software in block 212 determines that all components have the required data in all periods, then processing advances directly to a software block 221. Alternatively, if data are missing, then processing advances to a software block 213 where the user (20) is prompted by a missing financial data window (906) to provide the missing data or the location of the missing data. In some cases the user (20) may simply replace the null value with a zero. After the user (20) provides the missing data or the location of the missing data, the appropriate data tables (141-revenue, 142-expense and/or 143-capital) in the application database (50) are updated and processing advances to software block 221.

The next step in system processing is completed by software block 221 where the software in the block prompts the user (20) via an element of value definition data window

(907) to define the standard elements of value for each enterprise, to indicate if there will be sub-elements for the element and to identify the location of the data that will be used to quantify the period-to-period change for each element of value using pre-defined composite variables. If the calculation being run is a comparison calculation, then the user (20) is restricted to updating the previous specifications for new information. The standard elements of value with sample specifications are shown below in Table 16.

**Table 16**

<b>Standard Elements of Value</b>	<b>Sample Specification</b>	<b>Sub-Elements</b>	<b>Max Number</b>
Customers	Customer numbers: 1 - 21,877	Yes	10
Employees: Sales	All employees department 21	No	NA
Employees: Production	Job codes 17, 18, 19 and 33	No	NA
Employees: Support	Job codes 61 and 62	No	NA
Employees: Other	All job codes except 17, 18, 19, 33, 61 and 62, in all departments except 21	No	NA
Channel Partners	Customer numbers: 40,000 - 40,267	Yes	2
Vendors	Vendor numbers: 1 - 819	Yes	20
Production Equipment	ID number 40,000 - 49,999	Yes	5
Infrastructure	Only one per enterprise	No	NA
Brand names	Name(s)	Yes - 1 for each name	50*

\*Default system limit

After the information defining the standard elements of value has been stored in the element of value definition table (153), the user (20) is prompted to identify the source(s) of the data that will be used in computing the value of the pre-defined composite variables for the standard elements of value. Composite variables are numbers that are created by mathematically or logically combining transaction data, transaction ratios and/or transaction trends in any combination (i.e. transaction ratios and transaction trends, transaction ratios and transaction data, transaction ratios alone, etc.). For example, standard hours for a production worker could be defined as the actual production hours worked in a year multiplied by the average yearly ratio of actual hours to standard hours.

Table 17 shows the data typically required for calculation of the pre-defined composite variables for three standard elements of value.

**Table 17**

<b>Standard Elements of Value</b>	<b>Composite Variable Data by Item</b>
Customer	Average time between introduction of a new product and first purchase of the new product (AT), Days sales balance in Accounts Receivable (D), Months since first order (M); Months with orders from customer (MO), Months since last order (ML); Monthly invoice line item corrections (LIC); Monthly invoice line items (TLI), Monthly service calls (SC); Monthly sales in base currency (S); Monthly technical support calls (TC); Monthly new product order line items (NP), Monthly returned product quantity (RPQ), Monthly total product quantity (TPQ), Monthly repeat support calls (RTC); Monthly repeat service calls (RSC); Total monthly communications (TMC); New products as percentage of total products available (NPP), Weighted average days to pay (DA); Weighted average delivery variance ((actual delivery time X qty)/(promised delivery time X qty)) (DV); Weighted average percentage proprietary product/total product delivered (PP).
Employees: Production	Number of job/station certifications (JC); Cumulative employee suggestions (CS); Cumulative implemented suggestions (CIS); Cumulative training courses completed (CTC); Cumulative months employed (CM); Monthly production - standard hours (MSH); Monthly paid hours (MPH); Monthly rejected production quantity (MRQ); Monthly total production quantity (MPQ); Monthly pay including benefits (MP).
Brand names	Monthly average price premium/(discount) vs. industry average price (MPR), Monthly number of favorable mentions in trade press (MTP), Monthly number of hits on corporate web site (MWH), Monthly spending on advertising (MAD), Monthly average cost per 1,000 for advertising (ACT).

After the location of the composite variable data for each element of value has been stored in the composite variable location table (167) in the application database (50), processing advances to a software block 222.

The software in block 222 retrieves the variables required for calculating the composite variable for each element of value for each period and then stores the resulting information in the composite variable data table (168) in the application database (50) by item. After data storage is complete, system processing advances to a software block 223. The software in block 223 checks the composite variable data table (168) to determine if the required data are present for the time periods required for composite variable calculation in accordance with the date range previously calculated by the software in block 202 and stored in the system settings table (140). If data for all variables are present in all required time periods, then processing advances to a software block 225. Alternatively, if data for all of the required variables are not present, then processing advances to a software block 224. The software in block 224 prompts the user (20) via an enter missing data window (908) to provide the missing data required for composite variable calculation. When the user (20) has provided the required information, the new input is stored in the composite variable table (168) in the application database (50) and system processing returns to software block 223. If all data required for composite variable calculation is not present, then the process described previously is repeated. If all required data are present, then processing advances to software block 225.

The software in block 225 prompts the user (20) via a tax information data window (910) to provide an overall tax rate for the company and detailed schedules for federal income taxes plus any other taxes as shown in Table 18.



**Table 18**

<b>Tax</b>	<b>Example Schedule</b>
<b>Federal Income Tax</b>	15% of first \$250,000 in profit 25% of next \$500,000 in profit 35% of profit over \$750,000
<b>State Tax</b>	2.25% of revenue
<b>Overall Tax Rate</b>	33% of GAAP operating profit

After the information the user (20) provides is stored in the tax data table (173) in the application database (50), processing advances to a software block 226. The software in block 226 prompts the user (20) via an equity information data window (911) to provide historical and forecast (Fcst) information for each account included in the equity sub-component specification stored in the capital component definition table (152) as shown in Table 19.

**Table 19**

<b>Equity Account</b>	<b>Example Schedule</b>	<b>Actual /Fcst</b>
<b>301 - Preferred stock</b>	100,000 shares @ \$40/share 9/1/87 with yield 5% 250,000 shares @ \$90/share 3/31/98 with yield 8%	A F
<b>302 - Common Stock</b>	1,000,000 shares @ \$20/share on valuation date Price history for last 5 years	A A
<b>303 - Dividends</b>	Actual dividends last 5 years	A

After the information the user (20) provides is stored in the equity data table (144) in the application database (50), processing advances to a software block 227.

The software in block 227 prompts the user (20) via a liability information data window (912) to provide historical and forecast information concerning each account included in the financial liability sub-component stored in the capital component definition table (152) as shown in Table 20.

**Table 20**

<b>Liability Account</b>	<b>Example Schedule</b>	<b>Actual /Fcst</b>
201 - Accounts Payable	NA	
203 - Accrued Salary	NA	
205 - Short Term Debt	\$150,000 @ 12% annual, 12/31/91	A
	\$250,000 @ 11.7% annual, 3/17/93	A
	\$250,000 @ 11% annual, 6/30/99	F
215 - Long Term Debt	\$2,500,000 @ 8.5% annual, 9/1/93	A

After the information the user (20) provides is stored in the debt data table (174) in the application database (50), processing advances to a software block 228.

The software in block 228 calculates the current weighted average cost of capital using the information stored in the debt and equity tables (174 and 144, respectively) using Formula 1 shown below.

#### **Formula 1**

**Weighted average cost of capital =  $((D/V) \times R_D)(1-T) + (E/V \times R_E)$**

#### **Where:**

**D**= Value of Debt, **E**= Value of Equity, **R<sub>D</sub>**= Weighted Average Interest Rate of Debt, **T**= Tax Rate, **R<sub>E</sub>** = Rate of Return on Equity (based on historical information provided) and **V**=**(D+E)**

After the calculation is completed, processing advances to a software block 229. The software in block 229 compares the calculated value to the value previously specified by the user (20) in the system settings table (140). If the two values are different, then processing advances to a software block 230 which prompts the user via a cost of capital selection data window (913) to select the cost of capital figure to use for future calculations. The cost of capital specified by the user (20) is stored in the system settings table (140) and processing returns to block 229 and on to a software block 232. System processing passes directly to block 232 if the calculated and specified values of the cost of capital are identical.

The software in block 232 checks the asset liquidation price table (146) to determine if there are "current" (as defined previously) liquidation prices for all physical assets listed in the physical asset ID table (145). If there are "current" prices for all physical assets listed in the physical asset ID table (145), then processing advances to a software block 306 where the calculation of the composite variables begins. If, on the other hand, there are not "current" prices for all physical assets, then processing advances to a software block 235. The software in block 235 prompts the user (20) via a liquidation price entry data window (914) to provide liquidation prices for all physical assets that don't have "current" values. The user (20) is given the option of specifying a liquidation value as a fixed price, as a percentage of original purchase price or as a percentage of book value (as stored in the basic financial system database (10)). After the required information has been entered by the user (20) and stored in the asset liquidation price table (146) in the application database (50), system processing advances to block 306.

#### **CALCULATE COMPOSITE VARIABLES**

The flow diagram in FIG. 6 details the processing that is completed by the portion of the application software (300) that calculates the composite variables for each element and sub-element of value. Processing begins in software block 306. The software in block 306 checks the composite variable table (156) to determine if the composite variables for all elements of value have been calculated within the maximum allowable time period specified by the user (20) and stored the system settings table (140). As in the related U.S. patent 5,615,109, a calculation or sort completed within the acceptable time limit is defined as a "current" calculation or sort. If the examination of the composite variable table (156) reveals that the composite variables for all elements of value are "current", then processing advances to a software block 402. Alternatively, if some or all of the composite variables don't have current values, then processing advances to a software block 307.

The software in block 307 retrieves the definition for the next element of value from the element of value definition table (153) in the application database (50) and then uses the retrieved information together with the information in the composite variable location table (167) and the system settings table (140) to retrieve all the required data, by item, for the required months for use in calculating the composite variable for the element of

value being analyzed. Calculation of the composite variables is completed one item at a time before the individual item values are added together to calculate the composite variable for the element (or sub-element). The composite variables are calculated for each month required for element valuation in accordance with the formulas like those shown in table 21 for the three sample elements detailed in Table 17.

**Table 21**

Element of Value	Composite Variable Formulas*
Customers	$S \times (.05 \times (((TC - RTC)/TC) + ((TPQ - RPQ)/TPQ) + (1/DV) + ((SC - RSC)/SC) + ((TLI - LC)/TLI)) + (.125 \times ((ML/ML_{AVG}) + (D/DA))) + (.125 \times (((NP/TLI)/NPP) + (PP_{AVG}/PP)) + (.125 \times (((TMC_t - TMC_{(t-1)})/TMC_{(t-1)}) + MO/M)$
Employees: Production	$(MP) \times (.166 \times (MSH/MPH) + (MPQ-MRQ/MPQ) + ((CTC_t - CTC_{(t-12)})/12) + ((JC_t - JC_{(t-12)})/12) + ((CS_t - CS_{(t-12)})/12) + (CIS/CM))$
Brand names	$(MAD) \times (.25 \times (MPR + ((MTP_t - MTP_{(t-1)})/MTP_{(t-1)}) + ((MWH_t - MWH_{(t-1)})/MWH_{(t-1)}) + ((ACT_t - ACT_{(t-12)})/ACT_{(t-12)})))$
WHERE:	$X_t = \text{Value of X in period t}$ $X_{(t-1)} = \text{Value of X in period (t-1)}$ $X_{(t-12)} = \text{Value of X in period (t-12) - one year ago}$ $X_{AVG} = \text{Long term average value of X}$

\* see Table 17 for variable definitions

The calculated composite variables by item and element are stored in the composite variable table (156). The item level composite variable data are also stored at this time in the composite variable data table (168) in the application database (50) before processing advances to a software block 308.

The software in block 308 checks the element of value definition table (153) in the application database (50) to see if the user (20) has specified that there will be sub-elements of value for the element of value being analyzed. If the user (20) has indicated that there will be no sub-elements of value for this element, then processing returns to block 306. As described previously, if the software in block 306 determines that all elements of value have current composite variables, then processing will advance to

block 401. Alternatively, if there are elements of value without current composite variables, then processing returns to block 307 as described above. If the user (20) has instead specified that there will be sub-elements of value for the element of value being analyzed, then processing advances to a software block 309.

The software in block 309 checks the system settings table (140) to determine if the calculation being completed is a stand-alone calculation or a comparison to a prior calculation. If the software in block 309 determines that the current calculation is not being used for a comparison, then the processing advances to a software block 315. The software in block 315 retrieves the composite variable data by item for the element being analyzed from the composite variable data table (168) before creating a normalized set of composite variable data for each item within the element of value being analyzed. The normalized value for each composite variable data element for each item in each period is then calculated using Formula 2 shown below.

#### Formula 2

$$\text{Normalized Value} = \frac{\text{Current value} - \text{MN}}{(\text{MP} - \text{MN})}$$

**Where: MN = minimum positive or most negative data value for all element items**

**MP = maximum positive data value for all element items**

After the normalized data are saved in the normalized composite variable data table (169) in the application database (50), system processing advances to a software block 316. The software in block 316 uses an unsupervised “Kohonen” neural network that uses competitive learning to create a clustering scheme and segment the element of value. As shown in FIG. 10 a “Kohonen” network has only two layers - an input layer (712) that holds the input nodes (750-x) where the different inputs are sequentially entered. The input patterns are transmitted to an output layer (713) which has one node (760-x) for each possible output category. The input layer and the output layer are fully interconnected as shown in FIG. 10. The different variables are defined in Table 22.

**Table 22**

Variable	Definition
P	The number of items for the element. Equals the number of different patterns that will be presented to the network
M	The number of variables the in the composite variable for the element as well as the number of input nodes (750-1 through 750-M)
N	The maximum number of sub-elements for this element (default is 20) as well as the number of output nodes (760-1 through 760-N)
$\omega_{ij}$	Represents the connection strength between unit j of the input layer (712) and unit i of the output layer (713)
$X_j$	Represents the input vector which is the normalized value of the “j <sup>th</sup> ” item composite variables
$V_i$	Matching value - measures how closely the weights of a given node match the input vector

“Kohonen” network processing begins when the software in block 316 initializes at random the weights (716) between the output layer (713) and the input layer (712) with small values. In the next step the system starts sequentially entering the normalized composite variable data from the normalized composite variable data table (169) into the input layer (712). The normalized value for each variable is entered into a different input node (750-x) and transmitted from there to the output layer (713). The nodes in the output layer (760-x) each compute their matching values ( $V_i$ ) using Formula 3 shown below:

### Formula 3

$$\mathbf{v}_i = \sum (\omega_{ij} - \mathbf{x}_j)^2$$

The matching value ( $V_i$ ) essentially represents the distance between the vectors ( $\omega_i$ ) and  $x$ . Therefore, the output node (760) with the lowest matching value is also the node that most closely matches the input vector. The unit that is closest to the input is declared the winner and its weight ( $\omega_{ij}$ ) along with the weights of the neighboring output nodes are updated. The change in weight for the winning node and its neighbors is calculated using Formula 4 shown below.

**Formula 4**

$$\Delta \omega_{ij} = \alpha (X_j - \omega_{ij}) \quad \text{where: } \alpha \text{ represents the learning rate (see Formula 5)}$$

The application of this formula diminishes the difference between the weights of the output nodes and the weights of the input vectors. Output nodes that are not neighbors of the winning node are not updated. The output nodes are updated after each input and over time the application of the formulas shown above will tend to create clusters of similar nodes.

The input vectors (data patterns) are cycled through the "Kohonen" network a pre-determined number of times which are referred to as epochs. The total number of epochs (T) will be set by the software to somewhere between 500 and 10,000 depending upon the number of composite sort variables used for the element. The neighborhood size, that is the quantity of adjacent nodes that are considered to be neighbors, is adjusted downward from its initial value of 75% of the value of N by one node at a time as the number of epochs increases from zero (0) to its maximum number (T). The learning rate ( $\alpha$ ) is determined by Formula 5 shown below.

**Formula 5**

$$\alpha = 0.2 \times (1 - (T/10,000))$$

Once the Kohonen network processing has been completed for the specified number of epochs (T), the software in block 316 arbitrarily assigns a number to each output node (760-x). The software in block 316 then calculates the distance between the input vector (x) of each item and the weight in each output node (760-x) using Formula 3. The software in block 316 then assigns the number of the closest output node (760-x) to the item and stores the resulting information in the sub-element definition table (154) in the application database (50). The software in block 316 also stores the final value of all network weights in the sub-element weights table (157) in application database (50).

After the network weights and information assigning each item to a sub-element have been stored in the appropriate tables in the application database (50), processing advances to a software block 317. The software in block 317 retrieves the item data for each sub-element by month before calculating a composite variable for each sub-element

for each required time period using the appropriate formula (described previously) for the sub-element being analyzed. The results of these calculations are stored in the composite variable table (156) in the application database (50) before processing returns to block 306. As described previously, if the software in block 306 determines that all elements of value have current composite variables, then processing will advance to block 401. Alternatively, processing returns to block 307 as described previously.

If the software in block 309 determines that the calculation being completed is a comparison to a prior valuation, then processing advances to a software block 310. The software in block 310 retrieves the sub-element weights from the previous calculation from the sub-element weights table (157) and reestablishes the prior sub-element assignments by using Formula 3 to determine the appropriate sub-element for each item. When this processing has been completed, processing advances to a software block 312.

The software in block 312 checks the composite variable data table (168) to see if there are any new items for elements being analyzed. If there are no new items, then processing advances to block 317 and on to block 306 as described previously. Alternatively, if the software in block 312 determines that there are new items, then processing advances to a software block 313.

The software in block 313 determines the appropriate sub-element assignment for each new item by calculating the normalized value of the input vector for each new item and using formula 3 to determine which output node (i.e., which sub-element from the previous calculation) each item should be assigned to. The inputs for these calculations are stored in the normalized composite variable data table (169) and the results are stored in the composite variable data table (168) in the application database before processing advances to block 317 and on to block 306 as described previously.



## CALCULATE COMPONENTS OF VALUE

The flow diagram in FIG. 7 details the processing that is completed by the portion of the application software (400) that calculates the components and sub-components of value. Processing begins in a software block 402. The software in block 402 checks the enterprise value table (170) in the application database (50) to determine if there are "current" valuations for all enterprises for the date for which the current valuation is being calculated. If there are "current" valuations for all enterprises, then processing advances to a software block 415 where the software in the block calculates the total company current operation value. However, if some or all of the enterprises don't have "current" valuations, then processing advances to a software block 403.

The software in block 403 retrieves the definition for the next enterprise that doesn't have a "current" valuation from the enterprise definition table (155) in the application database (50). Processing then advances to a software block 404.

The software in block 404 checks the data from the revenue component definition table (150) for the revenue component of the enterprise being valued to determine if there is a "current" valuation for the component. If there is a "current" valuation for the revenue component, then processing advances to a software block 407 where the values of the expense component or expense sub-components for the enterprise are checked to determine if they are "current". However, if the revenue component valuation isn't "current", then processing advances to a software block 405. The software in block 405 retrieves the information for the revenue component from the revenue data table (141) and processing advances to a software block 406. In accordance with the present invention, the revenue component value is calculated for the specified date of valuation using Formula 6 shown below.

**Formula 6**

$$\text{Value} = F_{f1}/(1+K) + F_{f2}/(1+K)^2 + F_{f3}/(1+K)^3 + F_{f4}/(1+K)^4 + (F_{f4} \times (1+g))/((K-g) \times (1+K)^4)$$

Where:

$F_{fx}$	= Forecast revenue, expense or capital for year x after valuation date (from advanced financial system)
$K$	= Cost of capital - % per year (from system settings)
$g$	= Forecast growth rate to perpetuity - % per year (from advanced financial system)

After the valuation of the revenue component is complete, the result is stored in the revenue component definition table (150) in the application database (50) and processing advances to a software block 407.

The software in block 407 checks the expense component definition table (151) in the application database (50) to determine if there are "current" valuations for all expense components or sub-components in the enterprise being valued. If the user (20) has previously stored information in the system settings table (140) specifying a "simplified" analysis, then the expense component values will be checked. Alternatively, if the user (20) has not selected a simplified analysis, then the expense sub-component values will be checked. If there are "current" valuations for the expense components or all sub-components, then processing advances to a block 410 where the values of the capital components for the company are checked to determine if they are "current". However, if some or all of the expense components or sub-components don't have "current" valuations, then processing advances to a software block 408. The software in block 408 retrieves the information from the expense data table (142) for the expense component or the next expense sub-component that doesn't have a "current" valuation. Processing then advances to a software block 409. In accordance with the present invention the valuation of the expenses is calculated for the specified date of valuation using formula 6. After the valuation of the expense component or expense sub-component has been completed, the results are stored in the expense component definition table (151) in application database (50) and processing returns to a software block 407. If there are still expense sub-components that don't have current valuations, then the processing described above is repeated for the next sub-component. Alternatively, if the expense

component or all expense sub-components have current valuations, then processing advances to a software block 410.

The software in block 410 checks the capital component definition table (152) in the application database (50) to determine if there are "current" valuations for all capital components. If the user (20) has previously stored information in the system settings table (140) specifying a "simplified" analysis, then the capital component value for the enterprise will be checked. Alternatively, if the user (20) has not selected a simplified analysis, then the standard capital sub-components will be checked. If there are "current" valuations for all capital components, then processing advances to a software block 414 where the enterprise current operation value is calculated. If the valuation for the capital component or some of the capital sub-components is not "current", then processing advances to a software block 411. The software in block 411 retrieves the information required for valuation of the next capital component or sub-component that doesn't have a "current" valuation from the capital data table (143) in the application database (50). Processing then advances to a software block 412. The software in block 412 calculates the value of the capital component or capital sub-component using formula 6. After the valuation of the capital component or a capital sub-component is complete, the results are stored in the capital component definition table (152) in the application database (50) and system processing returns to block 410. If there are still capital sub-components that don't have current valuations, then the processing described above is repeated for the next sub-component. Alternatively, if the capital component or all capital sub-components have current valuations, then processing advances to a software block 414.

The software in block 414 calculates the current operation value of each enterprise by summing the previously calculated component and sub-component values as shown in Table 4. The calculated value for the enterprise current operation is stored in the enterprise value table (170) in the application database (50) and processing returns to block 402 which again checks the enterprise value table (170) in the application database (50) to determine if all enterprises have "current" values. If there are still enterprises without "current" values, then processing advances to block 403 and the processing described in the preceding paragraphs is repeated for another enterprise. Alternatively, if all the enterprises have "current" values, then processing transfers to a block 415 where the software in the block adds the enterprise values together to calculate the value of the

current-operation for the total company. The total company current-operation value is stored in the enterprise value table (170) in the application database (50) and processing advances to a software block 500 where the predictive model specification and optimization is started.

### **PREDICTIVE MODEL SPECIFICATION AND OPTIMIZATION**

The flow diagrams in FIG. 8A, FIG. 8B, FIG. 8C and FIG. 8D detail the processing that is completed by the portion of the application software (500) that uses predictive models to determine the relationship between the elements and sub-elements of value and the revenue, expense, capital and or cash flow of all defined enterprises. Processing in this section begins in a software block 501. The software in block 501 checks the system settings table (140) to see if the user (20) has specified a summary (cash flow) analysis in the system settings. If the user has selected a summary analysis, then processing advances to a software block 570. Alternatively, if the user has not specified a summary analysis, then processing advances to a software block 502.

The software in block 502 checks the revenue model weights table (159) in the application database (50) to determine if the revenue components for all enterprises have "current" models. If there are revenue components without "current" predictive models, then processing advances to a software block 503 where the information specifying the next revenue component is retrieved from the revenue component definition table (150) in the application database (50). After the revenue component definition has been retrieved, processing advances to a software block 504 where the software in the block creates a predictive time series neural net model for the revenue component. More specifically, the software in the block creates a neural network model that relates the elements and sub-elements of value for a given enterprise to the revenue component.

Neural networks are increasingly being used for statistically modeling the relationships between sets of data. One of the main reasons for the increase in their use is that they are effective in modeling relationships even when there are nonlinear relationships and interactions between independent variables. Neural networks consist of a number of processing elements (hereinafter, referred to as nodes) that send data to one another via connections. The strengths of the connections between the nodes are referred to as

weights. As shown in FIG. 9, there are three types of nodes, input nodes (710-x), hidden nodes (720-x) and output nodes (730). Input nodes receive data values from input variables (701). A threshold node (710-THRESH) is a special class of input node (710-x) with a constant weight of 1 on the connection to a hidden node (720-x). Hidden nodes (720-x) create intermediate representations of the relationship between input data and the output values. It is important to note that while the diagram in FIG. 9 shows only one layer of hidden nodes (703), in many cases a network model will contain several layers of hidden nodes. Finally, output nodes (730) produce output variables (705).

The action of a neural network is determined by two things: the architecture, that is how many input, hidden and output nodes it has; and the values of the weights. A neural network “learns” by modifying its weights (706 and 707) to minimize the difference between the calculated output value (705) and the actual output value. The difference between the calculated output value and the actual output value is defined as the error function for the network. For revenue components such as those specified by the software in block 504, the error function is defined by Formula 7.

#### Formula 7

$ERR(W)_k = 1/2 (R_k - Y(W))^2$	
Where:	
W	= a set of weight values
$ERR(W)_k$	= error function for W for period k
$R_k$	= actual/forecast revenue for period k
$Y(W)$	= output value for W

The process for minimizing the error function will be detailed after the specification of the network architecture is explained.

The software in block 504 determines the number of the input nodes and hidden nodes for each network as a function of the number of elements and sub-elements of value associated with the enterprise revenue component. There are also additional input nodes for prior period revenue and for a threshold node. For the system of the present invention, there is a minimum of twelve (12) input nodes and 13 hidden nodes for each predictive mode. The minimum number of input nodes is derived by adding one node for

each of the ten (10) standard elements of value (see Table 16) to the 2 extra nodes, for the threshold and prior period revenue. The minimum number of hidden nodes is derived by adding one (1) to the minimum number of input nodes. Table 23 shows the calculation of the number of nodes in the example predictive revenue model

**Table 23**

Standard elements of value	Sub-elements?	Maximum sub-elements	Actual total (sub)-elements
Customers	Yes	10	5
Employees: Sales	No	NA	1
Employees: Production	No	NA	1
Employees: Support	No	NA	1
Employees: Other	No	NA	1
Channel Partners	Yes	2	1
Vendors	Yes	20	4
Production Equipment	Yes	5	2
Infrastructure	No	NA	1
Brand names	Yes - 1 for each	50	2
<b>Subtotal Inputs:</b>			<b>19</b>
Threshold & Prior Period			2
<b>Total Input Nodes:</b>			<b>21</b>
Hidden Node Adder			1
<b>Total Hidden Nodes:</b>			<b>22</b>

The software in block 504 sets the initial number of hidden layers to one. The software in block 504 also establishes one output node for the revenue and sets all weights to random numbers between 0 and 1 (except the threshold node weight which is fixed at 1).

The processing completed by all of the network nodes (710-x, 720-x and 730) is similar. The input nodes (710-x) receive their input from system inputs while the hidden and output nodes (720-x and 730) receive input from other nodes. Each node multiplies the received input by the corresponding weight (706 or 707) to produce a weighted sum. The network applies a sigmoid or linear function to the weighted sum to determine the state of

the node. The state of each node is then passed on to the next layer along a weighted connection or it is used to generate an output variable. When the network architecture including the nodes has been specified by the software in block 504, then processing advances to a software block 525 where network optimization begins.

The normal operation of a neural network requires the use of very large amounts of data to train the network to minimize the error function and then test the networks predictive capabilities. The preferred embodiment of the present invention minimizes the need for very large data sets by using genetic algorithms to find the weights (W) that reduce the error function to an acceptable level before optimizing the network using the backpropagation algorithm to determine the “best fit”. The software in a block 525 uses genetic algorithms to find solutions for the current error minimization problem by evolving a set of solutions toward the desired goal of having an error function value of zero. More specifically, the genetic algorithms in block 525 create and maintain a population of the software equivalent of DNA chromosomes (hereinafter, chromosomes) that “evolve” toward the specified goal by using selective crossover and random mutation to generate new chromosomes. For this application, the chromosomes (see Table 24 below) encode the network weights.

**Table 24**

0	Gene 1
X	Gene 2
0	Gene 3
X	Gene 4
0	Gene 5

Each individual “gene” represents a weight between two sets of nodes. The fitness of each chromosome in the population is evaluated by the proximity of the resulting solution to the expected objective function maximum (the maximum of the objective function corresponds to the minimum error level of the neural network). Selective crossover in a genetic algorithm gives a preference to the chromosomes (sets of weights) that are the most fit (e.g., have lowest error and highest objective function outputs). Crossover is a form of reproduction that separates each of two individual chromosomes into two

separate pieces at a random break point. Crossover is completed when the algorithm recombines the top piece from the first chromosome with the bottom piece of the second chromosome and the bottom piece from the first chromosome with the top piece from the second chromosome to produce two new chromosomes that have a mix of "genes" from each of the original chromosomes. Giving a preference to the most fit chromosomes increases the likelihood that the new chromosomes will produce more fit solutions than the precursor chromosomes. Mutation is the random change in the value of a randomly selected "gene". Mutation occurs to "genes" during crossover. It also occurs in individual chromosomes within the population. When a population of chromosomes has been crossed over and mutated, a new generation of the population is created. The fitness of the chromosomes within the new population is evaluated and unless one of the chromosomes produces an acceptable solution (a solution where the error level is below the target), the process is repeated. Over time the selective crossover will increase the relative fitness of the population and decrease the difference between the best and worst chromosomes.

The evolutionary process is enhanced in the present invention using three separate mechanisms. First, the fitness measures for individual chromosomes are re-scaled before crossover by the software in block 525 whenever the difference between the fitness of the top 10% of population and the bottom 10% of the population is less than 5% of the expected solution. To accomplish this, the fitness of the chromosome(s) with the lowest fitness is arbitrarily changed to 10% of the target value and the fitness of the chromosome(s) with the highest fitness is set to 95% of the target value. The remaining chromosomes fitness values are adjusted accordingly. This adjustment has the effect of restoring the relative advantage that the fitter chromosomes have in being selected for crossover.

The second mechanism for speeding the evolutionary process is to pick only the fittest members of a population for inclusion in the next generation. For this procedure, the current generation is combined with the two preceding generations and the fittest third from the combined population is carried forward for crossover and mutation in the next generation by the software in block 525. Finally, the sensitivity of the solution to the inclusion of all "genes" is tested when the fitness of a chromosome reaches the target level or the fitness of the population fails to increase for the maximum number of



successive generations specified by the user (see System Settings, Table 12). The highest level of fitness achieved is established as the new target and processing advances to a block 530 after the resulting genes are stored in the revenue model genes table (158). The software in block 530 creates parallel populations where the “genes” (weights) associated with one element or sub-element of value are removed from each chromosome before processing advances to a software block 535.

The software in a block 535 repeats the evolution process using the parallel population with the highest initial average fitness. If the fitness level of a chromosome in the parallel populations exceeds the target value after a minimum number of generations (equal to the user specified maximum - see System Settings, Table 12) or the fitness of the population fails to increase for the user specified maximum number of successive generations, then processing advances to a block 540. If the software in block 540 determines that a chromosome in the parallel population has reached a new target level, then the genes are stored in the revenue model genes table (158) and the processing returns to a block 530 where process of creating parallel populations by removing element of value “genes” is repeated. The overall process of evolution and removing elements and sub-elements of value continues in this manner until the new parallel populations fail to reach a new target level and processing is then advanced to a block 545. The software in block 545 uses the chromosome that achieved the highest fitness to initialize a feed-forward neural network. In a manner that is well known, the network is then trained by the software in a block 550 using a traditional backpropagation algorithm to further minimize the error function associated with the network. The resulting weights for the enterprise are then saved in the revenue model weights table (159) in the application database (50) and processing returns to a block 502.

If the software in block 502 determines that there are “current” revenue models for all enterprises, then processing advances to a software block 505. The software in block 505 checks the expense model weights table (161) in the application database (50) to determine if the expense component or all expense sub-components have “current” models. If the user (20) has previously stored information in the system settings table (140) specifying a “simplified” analysis, then the expense component model will be checked before processing advances to a software block 507 or to a software block 511. Alternatively, if the user (20) has not selected a simplified analysis, then the standard

expense sub-component models will be checked before processing advances to block 507 or block 511. In either case, processing will advance to block 507 if the models aren't "current" and to block 511 if they are "current".

The software in block 507 retrieves the information specifying the expense component or the next expense sub-component from the expense component definition table (151) in the application database (50). After the required information is retrieved, processing advances to a block 508 where the predictive expense model is specified in a manner similar to that described previously for the predictive revenue model. From block 508, processing advances to blocks 525, 530, 535, 540, 545 and 550 where the genetic evolution of the fittest solution is completed in a manner similar to that described above for the predictive revenue model. As part of this processing expense model genes are stored in the expense model genes table (160) in a manner identical to that described previously for the storage of revenue model genes. If there are sub-components, then the process described above is repeated until all expense sub-components have "current" models. When all expense components or all expense sub-components have "current" models, processing advances to a software block 511.

The software in block 511 checks the capital model weights table (163) in the application database (50) to determine if the capital component or all capital sub-components have "current" models. If the user (20) has previously stored information in the system settings table (140) specifying a "simplified" analysis, then the capital component model will be checked before processing advances to a software block 513 or to a software block 601. Alternatively, if the user (20) has not selected a simplified analysis, then the standard capital sub-component models will be checked before processing advances to block 513 or block 601. In either case, processing will advance to block 513 if the models aren't "current" and to block 601 if they are "current".

The software in block 513 retrieves the information specifying the capital component or the next capital sub-component from the capital component definition table (152) in the application database (50). After the required information is retrieved, processing advances to a block 514 where the predictive capital model is specified in a manner similar to that described previously for the predictive revenue and expense models. From block 514, processing advances to blocks 525, 530, 535, 540, 545 and 550 where the

genetic evolution of the fittest solution is completed in a manner similar to that described above for the predictive revenue and expense model. As part of this processing capital model genes are stored in the capital model genes table (162) in a manner identical to that described previously for the storage of revenue and expense model genes. If there are sub-components, then the process described above is repeated until all capital sub-components have "current" models. When all capital components or all capital sub-components have "current" models, processing advances to a block 601 where valuations are calculated for the elements and sub-elements of value.

If the user (20) had specified a summary, cash flow analysis, then the processing described in the preceeding paragraphs is skipped when processing advances to software block 570 from block 501. The software in block 570 retrieves information from the revenue component definition table (150), the expense component definition table (151), the capital component definition table (152) and the enterprise definition table (155) as required to calculate the cash flow value for each enterprise and for the company as a whole. The software in block 570 uses the previously calculated revenue, expense and capital component and/or subcomponent values to calculate the value for for each enterprise and for the company as a whole and stores the results in the enterprise value table (170) before processing advances to a software block 571.

The software in block 571 checks the cash flow model weights table (175) in the application database (50) to determine if the all enterprises have a "current" cash flow model. In if all the models aren't "current", then processing advances to a block 572. If all models are current, then processing advances and to block 622 if they are "current".

The software in block 572 retrieves the information specifying the cash flow information for the next enterprise from the enterprise definition table (155) in the application database (50). After the required information is retrieved, processing advances to a block 573 where the predictive cash flow model is specified in a manner similar to that described previously for the predictive revenue and expense models. From block 573, processing advances to blocks 525, 530, 535, 540, 545 and 550 where the genetic evolution of the fittest solution is completed in a manner similar to that described above for the predictive revenue and expense model. As part of this processing cash flow model genes are stored in the cash flow model genes table (176) in a manner identical to

that described previously for the storage of revenue, expense and capital model genes. When all enterprises have "current" cash flow models, processing advances to a block 622 where valuations are calculated for the elements and sub-elements of value.

### **VALUE ALL ELEMENTS AND SUB-ELEMENTS OF VALUE**

The flow diagram in FIG. 11A and FIG. 11B details the processing that is completed by the portion of the application software (600) that values all elements and sub-elements of current-operation value for all enterprises. If summary processing is being completed, then processing in this portion of the application software begins in a software block 622. Alternatively, if detailed processing is being completed, then processing begins in a software block 602. The software in block 602 checks the revenue component percentage table (164) in the application database (50) to determine if the revenue component models for all enterprises have "current" percentages. If there are revenue components without "current" percentages, then processing advances to a block 603 where the information specifying the next revenue component is retrieved from the revenue component definition table (150) and the revenue model weights table (159) in the application database (50).

After the revenue component information is retrieved, processing advances to a block 604 where relationships between the elements and sub-elements of value and the revenue component are determined. The software in block 604 uses the network weights (706 and 707) previously stored in the revenue model weights table (159) to segregate the hidden-layer (703) to output-layer (704) connection weights (707) for each hidden node (720-x) into the components associated with each input node (710-x). The portion of the output attributable to each input node is then determined by Formula 8 (shown below).

## Formula 8

$$\left( \sum_{k=1}^{k=m} \sum_{j=1}^{j=n} I_{jk} \times O_k / \sum_{j=1}^{j=n} I_{ik} \right) / \sum_{k=1}^{k=m} \sum_{j=1}^{j=n} I_{jk} \times O_k$$

Where

- $I_{jk}$  = Absolute value of the input weight (706) from input node j to hidden node k
- $O_k$  = Absolute value of output weight (707) from hidden node k
- m = number of hidden nodes
- n = number of input nodes

After the equation shown above is solved by the software in block 604, the portion of the revenue value attributable to each element or sub-element of value is calculated and stored in the revenue component percentage table (164) in the application database (50). The portion of the revenue value that can't be attributed to an element or sub-element of value is generally the portion that is attributed to the prior period revenue. This portion of the value will be referred to as going concern revenue component. After the storage of the revenue component percentages has been completed, processing returns to block 602. The software in block 602 checks the application database (50) to determine if all revenue components have "current" model percentages. If there are still revenue components without "current" percentages, then the system repeats the processing described in the preceding paragraphs. Alternatively, if all of the revenue component models have "current" percentages, then processing advances to a software block 605.

The software in block 605 checks the expense component percentage table (165) in the application database (50) to determine if all expense component or sub-component models for all enterprises have "current" percentages. If the user (20) has previously stored information in the system settings table (140) specifying a "simplified" analysis, then the expense component percentages will be checked. Alternatively, if the user (20) has not selected a simplified analysis, then the standard expense sub-component percentages will be checked. If there are expense components or sub-components without "current" percentages, then processing advances to a software block 606 where the information specifying the next expense component or sub-component is retrieved from the expense component definition table (151) and the expense model weights table (161) in the application database (50). After the expense component or sub-component information is retrieved, processing advances to a software block 607 where the

percentages of value for the expense component or sub-component are calculated in a manner identical to that described previously for revenue components. The portion of the expense value that can't be attributed to an element or sub-element of value is generally the portion that is attributed to the prior period expense. This portion of the value will be referred to as going concern expense component. After the storage of the percentages of the expense component or sub-component to the expense component percentage table (165) has been completed, processing returns to block 605. The software in block 605 checks the expense component percentage table (165) in the application database (50) to determine if all expense component or sub-component models have "current" percentages. If there are still expense component or sub-component models without "current" percentages, then the system repeats the processing described above. Alternatively, if all of the expense component or sub-component models have "current" percentages, then processing advances to a software block 608.

The software in block 608 checks the capital component percentage table (166) in the application database (50) to determine if all capital component or sub-component models for all enterprises have "current" percentages. If the user (20) has previously stored information in the system settings table (140) specifying a "simplified" analysis, then the capital component percentages will be checked. Alternatively, if the user (20) has not selected a simplified analysis, then the capital sub-component percentages will be checked. If there are capital component or sub-component models without "current" percentages, then processing advances to a software block 609 where the information specifying the next capital component or sub-component is retrieved from the capital component definition table (152) and the capital model weights table (163) in the application database (50). After the capital component or sub-component information is retrieved, processing advances to a software block 610 where the percentages of value for the capital component or sub-component are calculated in a manner identical to that described previously for revenue and expense components. The portion of the capital value that can't be attributed to an element or sub-element of value is generally the portion that is attributed to the prior period capital requirements. This portion of the value will be referred to as going concern capital value. After the storage of the percentages of the capital component or sub-component to the capital component percentage table (166) has been completed, processing returns to block 608. The software in block 608 checks the capital component percentage table (166) in the application database (50) to

determine if all capital components or sub-components have "current" percentages. If there are still capital component or sub-component models without "current" percentages, then the system repeats the processing described above (609 and 610). Alternatively, if all of the capital components or sub-components have "current" percentages, then processing advances to a software block 611.

The software in block 611 combines all the revenue component, expense component or sub-component and capital component or sub-component values together to calculate the overall value for each element or sub-element of value by enterprise as shown in Table 4. As part of the processing in this block, the calculated value of production equipment element (or sub-elements) of value is compared to the liquidation value for the equipment in the element. The stored value for the element (or sub-elements) will be the higher of liquidation value or calculated value. After the calculations are completed, processing advances to a software block 612 where the residual going concern value is calculated using Formula 9.

### Formula 9

$$\text{Residual Going Concern Value} = \text{Total Current-Operation Value} - \sum \text{Financial Asset Values} - \sum \text{Elements of Value} - \sum \text{Sub-Elements of Value}$$

After the residual going concern value is calculated for each enterprise, the values calculated for each element and sub-element of value (including going concern value) by the software in blocks 611 and 612 are stored by enterprise in the enterprise value table (170) in the application database (50). System processing then advances to a software block 772 where the preparation of the management reports is started.

If a summary (cash flow) analysis is being completed, then processing in this portion of the application software (600) begins in software block 622. The software in block 622 checks the cash flow percentage table (177) in the application database (50) to determine if all cash flow models for all enterprises have "current" percentages. If there are enterprise cash flow models without "current" percentages, then processing advances to a software block 623 where the information specifying the next enterprise cash flow is retrieved from the enterprise definition table (155) and the cash flow model weights table (175) in the application database (50). After the cash flow definition information is

The software in block 625 calculates the overall value for each element or sub-element by multiplying the cash flow percentages and the cash flow values. As part of the processing in this block, the calculated value of production equipment element (or sub-elements) of value is compared to the liquidation value for the equipment in the element. The stored value for the element (or sub-elements) will be the higher of liquidation value or calculated value. After the calculations are completed, the residual going concern value is calculated using Formula 9. After the residual going concern value is calculated for each enterprise, the calculated values for each element of value, each sub-element of value (including going concern value) and residual value are stored by enterprise in the enterprise value table (170) in the application database (50). System processing then advances to a software block 772 where the preparation of the management reports is started.

The flow diagram in FIG. 12 details the processing that is completed by the portion of the application software (700) that creates, displays and optionally prints financial management reports. The primary management report, the Operational Value Map™, report summarizes information about the elements and sub-elements of business value on the valuation date. If a comparison calculation has been completed, a Operational



Value Creation report can be generated to highlight changes in the elements and sub-elements of business value during the period between the prior valuation and the current valuation date.

System processing in this portion of the application software (700) begins in block 772. At this point in system processing, virtually all of the information required to produce the Value Map™ report has been calculated using the methods outlined in Table 1 as detailed in the preceding sections. As a result, the only computation that needs to be made is the calculation of economic equity. The software in block 772 retrieves the required information from the enterprise value table (170), debt data table (174) and equity data table (144) in the application database (50) and then calculates the economic equity for the business as a whole using Formula 10 (shown below).

### Formula 10

$$\text{Economic Equity} = (\text{Current Operation Value}) - (\text{Current Liabilities}) - (\text{Current Debt}) - (\text{Book* Equity Value})$$

\*calculated in accordance with GAAP

An equity value for each enterprise is then calculated by dividing the combined book and economic equity as required to balance the Value Map™ report totals in accordance with Formula 11 (shown below).

### Formula 11

$$\text{Enterprise Equity} = (\text{Current Enterprise Operation Value}) - (\text{Current Enterprise Liabilities}) - (\text{Current Enterprise Debt})$$

$$\text{where } \Sigma (\text{Enterprise Equity}) = \text{Book* Equity} + \text{Economic Equity}$$

\*calculated in accordance with GAAP

After the economic equity value and the enterprise equity values are calculated and stored in the economic equity values table (171), a summary Operational Value Map™ report for the entire company is created and stored in the reports table (172) and processing advances to a software block 773. The software in block 773 checks the system settings table (140) to determine if the current valuation is being compared to a previous valuation. If the current valuation is not being compared to a previous valuation, then processing advances to a software block 775. Alternatively, if the current valuation

is being compared to a previously calculated valuation, then processing advances to a software block 774.

The software in block 774 calculates Operational Value Creation Statements for each enterprise and for the business as a whole for the specified time period. After the Operational Value Creation Statements are stored in the reports table (172) in the application database (50), processing advances to a software block 775. The software in block 775 displays the summary Value Map™ report to the user (20) via a report data window (909).

After displaying the summary Value Map™ report, system processing advances to a software block 776 where the user is prompted via a report selection data window (915) to designate additional reports for creation, display and/or printing. The report selection data window (915) also gives the user (20) the option of having a report created to analyze the relationship between the market value of the business and the calculated business value. The user (20) has the option of creating, displaying or printing the Operational Value Map™ report for the company as a whole and/or for any combination of the enterprises within the company. The user (20) can also choose to create, display or print an Operational Value Creation Statement for the business as a whole and/or for any combination of enterprises if a comparison calculation were being made. The software in block 776 creates and displays all Operational Value Map™ reports and Operational Value Creation Statements requested by the user (20) via the report selection data window (915). After the user (20) has completed the review of displayed reports and the input regarding equity analysis and reports to print has been stored in the reports table (172), processing advances to a software block 777. The software in block 777 transfers processing to a software block 778 if the user (20) has chosen to have the relationship between market value and calculated business value examined. The software in block 778 compares the market value of the business to the calculated value by completing the Formula 12 for each complete valuation stored in the reports table (172).

**Formula 12**

$$((\sum E \times N) - D) = (Y \times V)$$

Where:

- E = Market price of equity for valuation date
- N = Number of shares of equity outstanding on valuation date
- D = Market value of debt on valuation date
- Y = Market value/calculated business value ratio
- V = Total calculated business value on the valuation date

The average ratio of market value to calculated business value and the standard deviation of the ratio are then calculated using standard regression analysis methods and stored in the equity forecast table (148) in the application database.

If the date of the current valuation is more than 60 days after the current system date, then the software in block 778 will calculate a range for equity prices on the valuation date by combining the results of previous calculations of the relationship between equity value and calculated value with the forecast of future value that was just completed. The software will calculate the future equity value range using both the average ratio of total business value to total market value. The software in block 778 then prepares a report summarizing the results of the preceding calculations that is stored in the reports table (172) in the application database (50) and processing advances to a software block 779. If the user (20) elects not to complete the calculated valuation versus equity price analysis, then the software in block 777 advances processing directly to a software block 779.

The software in block 779 checks the reports tables (172) to determine if any reports have been designated for printing. If reports have been designated for printing, then processing advances to a block 780 which sends the designated reports to the printer (118). After the reports have been sent to the printer (118), processing advances to a software block 781 where processing stops. If no reports were designated for printing then processing advances directly from block 779 to 781 where processing stops.

Thus, the reader will see that the system and method described above transforms extracted transaction data and information into detailed valuations for specific elements of

a business enterprise. The level of detail contained in the business valuations allows users of the system to monitor and manage efforts to improve the value of the business in a manner that is superior to that available to users of traditional accounting systems and business valuation reports. The user also has the option of examining the relationship between the calculated business value and the market price of equity for the business.

While the above description contains many specificity's, these should not be construed as limitations on the scope of the invention, but rather as an exemplification of one preferred embodiment thereof. Accordingly, the scope of the invention should be determined not by the embodiment illustrated, but by the appended claims and their legal equivalents.

09761670-01B01